MPI-IO/Gfarmのご紹介と現在の開発状況

鷹津冬将

目次

- MPI-IO/Gfarm概要
- MPI-IO/Gfarmの開発状況
- MVAPICH2向けMPI-IO/Gfarm
- MPI-IO/Gfarmの使い方
- かんたんなサンプルプログラムと動作確認の方法
- 既知の不具合
- まとめと今後の展望

MPI-IO/Gfarmの概要

MPI-IO/Gfarmの概要

- MPI-IO/Gfarm[木村ら, 2011]は、Gfarm向けのMPI-IOの実装
- 複数のMPIプロセスからの同一ファイルへのアクセスをGfarm上の複数のファイルに分散させることでアクセス性能を向上させる
- MPICH2向けのものがgfarm_mpiioとして配布中

木村浩希, 建部修見. "MPI-IO/Gfarm: 分散ファイルシステム Gfarm のための MPI-IO の実装と評価." *情報処理学会論文誌* 52.12 (2011): 3239-3250.

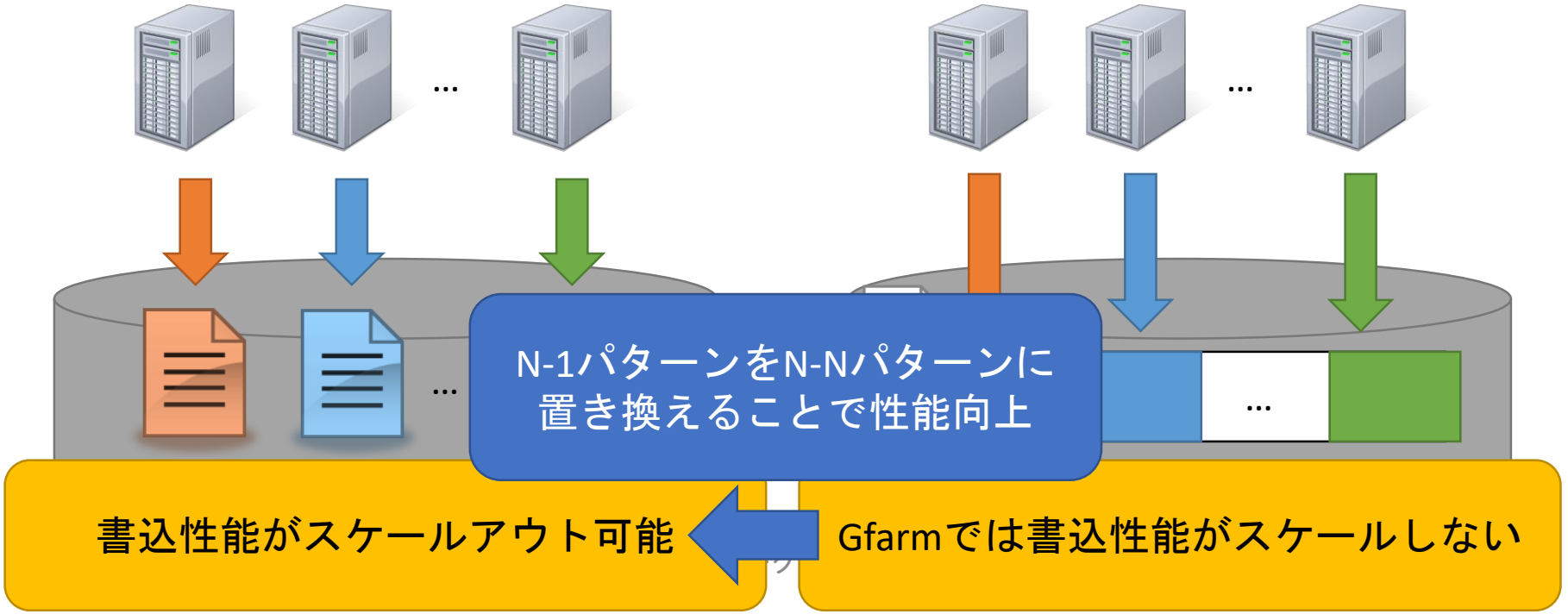
並列アクセスパターン

N-Nアクセスパターン

- 複数のプロセスが**それぞれ異なるファイル**に書き込む

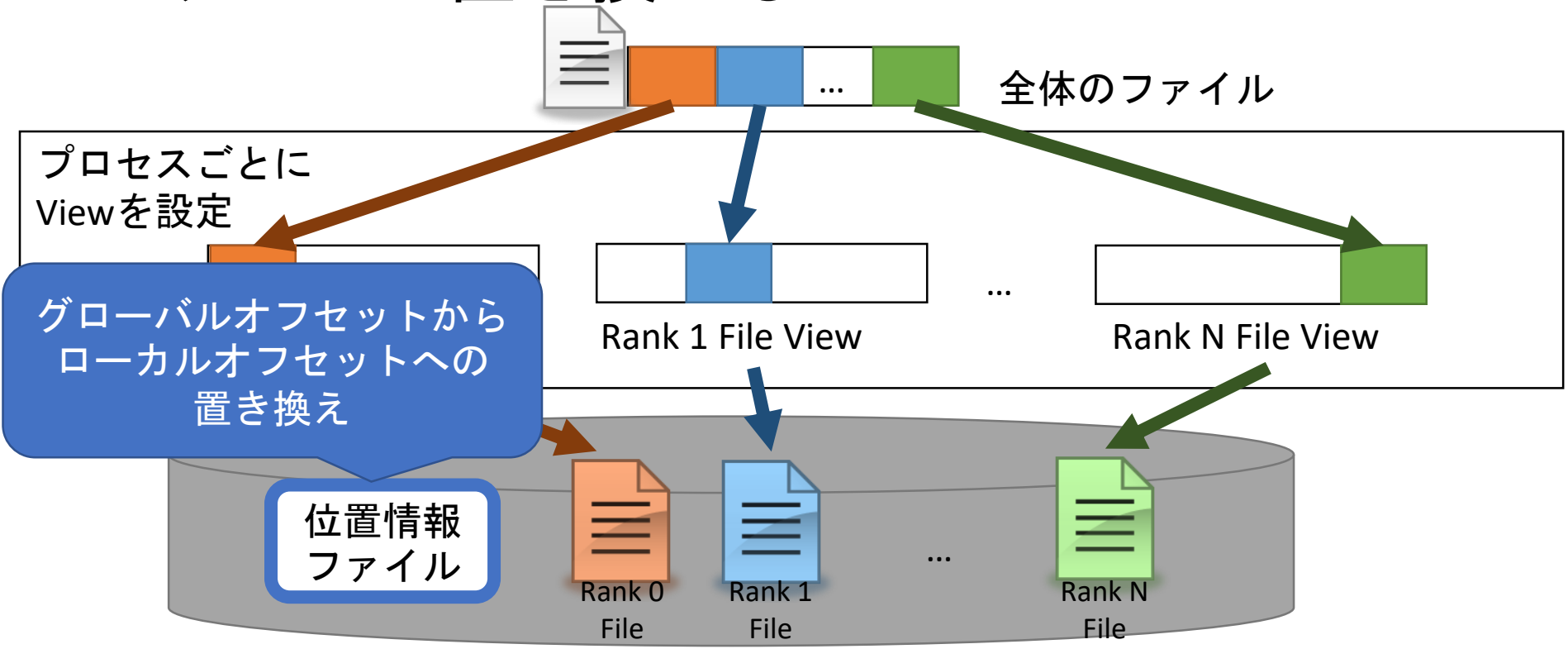
N-1アクセスパターン

- 複数のプロセスが**同一ファイル**に書き込む



MPI-IO/Gfarm

- N-1アクセスパターンをN-Nアクセスパターンに置き換える



MPI-IO/Gfarmの開発状況

MPI-IO/Gfarmの開発状況

- MPICH2向け
  - mpich2-1.3.2p1向け
  - Gfarm用のコードをROMIOに追加
  - Ver. 0.0.1としてリリース中
  - https://sourceforge.net/projects/gfarm/files/gfarm_mpiio/0.0.1/
- MVAPICH2向け
  - MPICH2向けのもの（Ver. 0.0.1）ベース
  - 現在開発中
  - 機能はMPICH2向けのものと変わらない

MVAPICH2向け MPI-IO/Gfarm

利用までの流れ

1. ダウンロードとパッチの適用
   - MVAPICH2(or MPICH2)をダウンロードし、gfarm_mpiioプラグインを利用可能にする
2. configureスクリプトの実行
   - 利用可能なファイルシステムとしてGfarmを選択
3. ビルドとインストール

ダウンロードとパッチの適用

```
# MVAPICH2-2.2のダウンロードと展開
$ wget http://mvapich.cse.ohio-state.edu/download/mvapich/mv2/mvapich2-2.2.tar.gz
$ tar zxvf mvapich2-2.2.tar.gz

# ad_gfarm ディレクトリのコピー
$ cp mvapich2/Makefile.mk ad_gfarm
$ cp -r ad_gfarm mvapich2-2.2/src/mpi/romio/adio/

# パッチの適用
$ patch -p0 < mvapich2/mvapich2.patch
$ cd mvapich2-2.2
$ ./autogen.sh
```

現在対応しているバージョン

必ず行うこと, autotoolsのバージョンが古い場合は最新版を入れる

configure スクリプトの実行

```
# 環境変数の設定
$ export CFLAG=“-I$GFARM_PREFIX/include’
$ export LIBS="-lgfarm"
$ export LDFLAGS="-L$GFARM_PREFIX/lib“


# configure スクリプトの実行
$ ./configure --enable-romio --with-file-system=ufs+nfs+gfarm
```

$GFARM_PREFIX
はGfarmをインストール
したパス

Gfarmを選択する。
nfs ufs pfs pvfs pvfs2 testfs xfs panfs gridftp
lustre gpfs zoidfs hfs piofs sfs
などが利用可能(“+”でつなぐ)

ビルドとインストール

```
# ビルド
$ make

# インストール
$ make install

# 組み込まれているかの確認
$ which mpirun
-> インストールしたmvapich2が出てくることを確認する

$ ldd `which mpirun` | grep gfarm
-> libgfarmがリンクされていることを確認する
```

MPI-IO/Gfarmの使い方

使い方

MPI_File_open() でファイルを指定する際にパスの先頭に**gfarm:**をつける

元のコード

```
   MPI_File_open(MPI_COMM_WORLD, “filename",
        MPI_MODE_CREATE | MPI_MODE_WRONLY, MPI_INFO_NULL,
&fh);
```

Gfarmに対応したコード

```
   MPI_File_open(MPI_COMM_WORLD, "gfarm:filename",
        MPI_MODE_CREATE | MPI_MODE_WRONLY, MPI_INFO_NULL,
&fh);
```

サンプルプログラムと動作サンプル

サンプルプログラムの概要

- 各プロセスは適当な値を同一ファイルに書き込む
- 4ノード・4プロセスで実行
- プログラムから見えるファイル
  - gfarm:testfile
- 実際に作成されるファイル
  - testfile
    - testfile/data/[0-3]-0
    - testfile/meta/[0-3]
    - …

サンプルプログラム(1/2)

```
 1 #include <unistd.h>
 2 #include <stdio.h>
 3 #include <stdlib.h>
 4 #include <stddef.h>
 5 #include <mpi.h>
 6
 7 int main(int argc, char *argv[]) {
 8     int rank;
 9     MPI_File fh;
10
11     MPI_Init(&argc, &argv);
12     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
13
14     MPI_Datatype new_type;
15     MPI_Type_vector(2, 1, 2, MPI_INT, &new_type);
16     MPI_Type_commit(&new_type);
```

ギャップがある
データ型を
定義

サンプルプログラム(2/2)

```
17
18    int buf[] = {rank + 1, rank + 2};
19
20    MPI_File_open(MPI_COMM_WORLD, "gfarm:testfile",
21        MPI_MODE_CREATE | MPI_MODE_RDWR, MPI_INFO_NULL,
&fh);
22
23    MPI_File_set_view(fh, rank * sizeof(int) * 3,
24        MPI_INT, new_type, "native", MPI_INFO_
25
26    MPI_File_write(fh, &buf, 1, new_type, MPI_STATUS_IGNORE);
27
28    MPI_File_close(&fh);
29    MPI_Finalize();
30
31    return 0;
32 }
```

ファイルの Open

Gfarmを指定

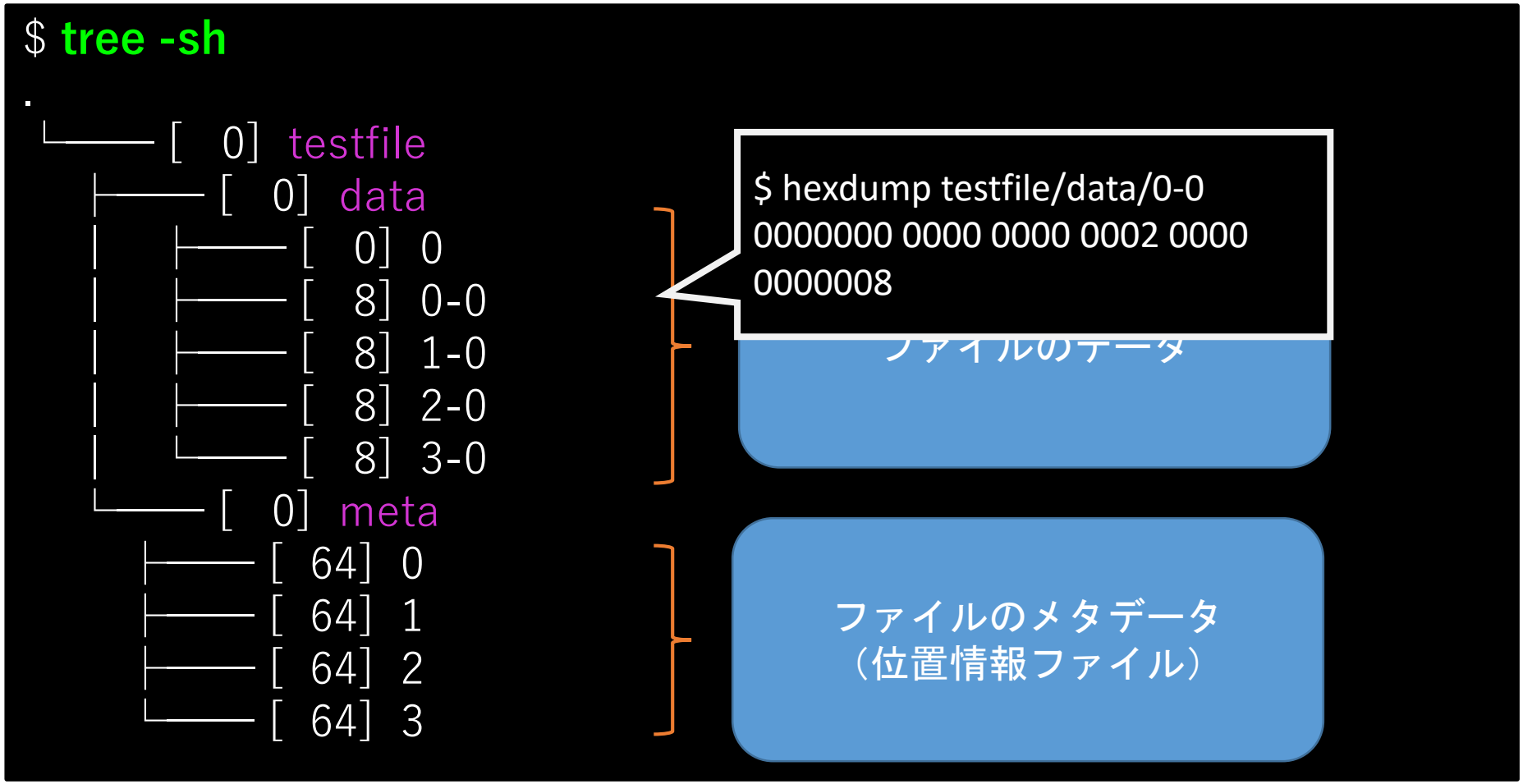
コンパイルと実行

```
# コンパイル
$ CC=mpicc make mpiio


# 確認
$ ls –l mpiio
-rwxrwxr-x 1 fuyumasa fuyumasa 7.9K Feb 22 20:53 mpiio*
$ ldd mpiio | grep gfarm
libgfarm.so.1 => <Gfarmをインストールしたパス>/lib/libgfarm.so.1


# 実行
$ mpirun –hostfile hostfile ./mpiio
```

ファイルが存在する

libgfarmが
リンクされている

ファイルの見え方


$ tree -sh
.
└── [0] testfile
    ├── [0] data
    │   ├── [0] 0
    │   ├── [8] 0-0
    │   ├── [8] 1-0
    │   ├── [8] 2-0
    │   └── [8] 3-0
    └── [0] meta
        ├── [64] 0
        ├── [64] 1
        ├── [64] 2
        └── [64] 3
$ hexdump testfile/data/0-0
0000000 0000 0000 0002 0000
0000008
ファイルのデータ
ファイルのメタデータ
（位置情報ファイル）

既知の不具合

- 読み込み出来ない
  - 書き込みは可能
  - MPICH2向けのものでも発生

  →現在調査中

まとめと今後の展望

- MPI-IOを利用したプログラムからGfarmの特徴を活用するためのプラグイン gfarm_mpiioのご紹介

- 今後の展望
  - バグ対応
  - OpenMPI 3.0対応